

Dokumentacja Projektowa – Gamifikacja Systemów PŁ

Skład zespołu:

- Bartłomiej Raj 251210
- Marcel Podlecki 251204
- Bartłomiej Jedyk 252795
- Wojciech Stochmiątek 251226
- Marcin Świstak 245944

Tytuł projektu:

- „Wyścig po Zaliczenie” (WpZ)

Zadanie 1 – Budowanie bazy wiedzy o projekcie

Na podstawie wstępnych analiz zespołu dotyczących doświadczeń studentów z systemami informatycznymi Politechniki Łódzkiej (zebranych m.in. podczas przygotowywania materiałów na temat "Czy WIKAMP musi być nudny?"), zidentyfikowano kluczowe platformy, procesy, wyzwania oraz potencjalne obszary do usprawnień. Ta wspólna baza wiedzy stanowi fundament dalszych prac projektowych:

1. Główne Systemy PŁ i Interakcje Studenckie:

Wikamp: Centralny system e-learningowy, służący do dostępu do materiałów dydaktycznych, zadań, ocen cząstkowych, ogłoszeń, forów dyskusyjnych oraz informacji o prowadzących. Jest intensywnie wykorzystywany przez studentów do nauki i komunikacji związanej z kursami, zarówno w domu, na uczelni, jak i w podróży. Zidentyfikowano go jako kluczowe źródło wiedzy przedmiotowej.

WebDziekanat: System administracyjny używany do zapisów na przedmioty, sprawdzania ocen końcowych, monitorowania statusu semestru i załatwiania spraw formalnych. Zwrócono uwagę na potencjalne trudności w nawigacji i nieintuicyjność interfejsu użytkownika (UI).

Dante: Platforma do oddawania i weryfikacji zadań programistycznych, w tym kontroli antyplagiatowej. Wskazano na potencjalnie długi czas oczekiwania na wyniki weryfikacji jako możliwe źródło frustracji.

System Biblioteczny: Umożliwia wyszukiwanie, rezerwację i zarządzanie wypożyczeniami zasobów bibliotecznych.

Mapa PŁ (NAVPL): Narzędzie do nawigacji po kampusie. Zauważono brak pewnych funkcjonalności, jak widok 3D budynków.

Inne Procesy/Systemy: Studenci wchodzą w interakcje również z systemami zapisów na WF/CJ, systemami obsługi miejsc w akademikach i na parkingach, procesami uzyskiwania zaświadczeń, systemami informacyjnymi o stypendiach oraz procesem obsługi e-legitymacji.

2. Typowe Zadania i Potrzeby Studentów (szczególnie na początku studiów):

Realizacja obowiązkowych szkoleń (BHP, biblioteczne).

Proces zapisów na przedmioty i konieczność szybkiego opanowania obsługi kluczowych systemów (Wikamp, WebDziekanat).

Potrzeba łatwego odnajdywania sal zajęciowych na rozległym kampusie.

Rejestracja i zrozumienie zasad działania systemu Dante dla zadań programistycznych.

Organizacja spraw administracyjnych i bytowych (akademik, parking, zaświadczenia).

Częste wyszukiwanie kluczowych informacji: szczegółowe plany zajęć, znaczenie skrótów, zasady oceniania, terminy, informacje o stypendiach, kontakty do jednostek uczelni, status tygodnia (parzysty/nieparzysty).

Efektywna nauka i powtarzanie materiału z wielu, często nowych, przedmiotów.

3. Identyfikowane Wyzwania i Potencjalne Źródła Frustracji:

Przeciążenie informacyjne i mnogość nowych systemów, zwłaszcza na pierwszym roku.

Trudności w szybkim dotarciu do specyficznych, potrzebnych informacji w rozproszonych systemach.

Nieintuicyjność i niska użyteczność niektórych interfejsów (szczególnie WebDziekanat).

Stres związany z presją czasu, zaliczeniami, kolokwiami i egzaminami.

Monotonia i brak zaangażowania w tradycyjnych metodach powtarzania materiału.

Konieczność szybkiego reagowania na zmiany (np. odwołane zajęcia, wyniki) i potrzeba łatwego dostępu do aktualnych danych.

Zróżnicowane warunki korzystania z systemów (hałas w akademiku/autobusie, ograniczony czas między zajęciami, różne warunki oświetleniowe) wpływające na efektywność.

4. Sugerowane Kierunki Usprawnień (wynikające z analizy):

Wprowadzenie zautomatyzowanych asystentów (chatbotów) do odpowiadania na często zadawane pytania administracyjne i nawigacyjne.

Umożliwienie interakcji z systemami w języku naturalnym.

Wykorzystanie AI do personalizacji ścieżek nauki i rekomendacji materiałów.

Stworzenie dedykowanych aplikacji mobilnych ułatwiających dostęp do kluczowych funkcji (np. dynamiczny plan zajęć).

Opracowanie narzędzi wspierających organizację studiów (np. współdzielony kalendarz terminów zaliczeń).

Automatyzacja procesów administracyjnych (np. obsługa e-legitymacji).

Zgromadzona wiedza wskazuje na duży potencjał w zakresie poprawy doświadczeń studentów, zwłaszcza poprzez zwiększenie zaangażowania w proces nauki i ułatwienie nawigacji po ekosystemie informatycznym PŁ. Gamifikacja jawi się jako obiecujące podejście do adresowania zidentyfikowanych wyzwań, szczególnie monotonii nauki i potrzeby motywacji.

Zadanie 2 – Burza mózgów i Wybór Obszaru Gamifikacji

Wykorzystując bazę wiedzy zgromadzoną w Zadaniu 1, zespół przeprowadził sesję burzy mózgów w celu identyfikacji procesów i działań w systemach PŁ, które najlepiej nadają się do wzbogacenia poprzez mechanizmy gamifikacyjne. Celem było znalezienie obszaru, gdzie gra mogłaby realnie zwiększyć zaangażowanie studentów i uatrakcyjnić często monotonne, lecz konieczne, aspekty studiowania.

Podczas dyskusji rozważano różne możliwości, mapując zidentyfikowane problemy i potrzeby studentów na potencjalne mechaniki gier:

Gamifikacja procesu onboardingu: Gra wprowadzająca nowych studentów w arkana systemów PŁ, procedur i lokalizacji na kampusie (adresowałaby problemy z natłokiem informacji i odnajdywaniem się).

Gamifikacja procesu administracyjnego: Gra symulująca załatwianie spraw w WebDziekanacie czy systemie stypendialnym (mogłaby potencjalnie oswoić studentów z tymi systemami, ale ma ograniczony potencjał angażujący).

Gamifikacja nawigacji po kampusie: Rozszerzona wersja Mapy PŁ z elementami gry eksploracyjnej (ciekawe, ale może być trudne technicznie i mniej związane z głównym celem studiów – nauką).

Gamifikacja procesu nauki i weryfikacji wiedzy: Gra wykorzystująca materiały i quizy z Wikampa do rywalizacji i utrwalania wiedzy (bezpośrednio adresuje problem monotonii nauki, stresu przed zaliczeniami i wykorzystuje istniejące zasoby).

Po przeanalizowaniu potencjalnych korzyści i złożoności implementacji, zespół doszedł do wniosku, że największy potencjał tkwi w gamifikacji procesu nauki i powtarzania materiału, ściśle powiązanego z systemem Wikamp. Taki wybór pozwala na:

1. Bezpośrednie wsparcie kluczowego celu studiów: Utrwalanie wiedzy i przygotowanie do zaliczeń.
2. Wykorzystanie istniejących zasobów: Możliwość (przynajmniej koncepcyjna) czerpania pytań i materiałów z Wikampa.
3. Adresowanie problemu monotonii: Zastąpienie lub uzupełnienie tradycyjnych metod nauki angażującą rozgrywką.
4. Wprowadzenie elementu społecznościowego: Możliwość rywalizacji i wspólnej zabawy ze znajomymi.

5. Motywowanie poprzez postęp i osiągnięcia: Wprowadzenie punktacji, rankingów, odznak.

Ostateczny Wybór Obszaru do Gamifikacji: Zespół jednogłośnie zdecydował się na opracowanie formy gamifikacyjnej koncentrującej się na utrwalaniu wiedzy z wybranych przedmiotów akademickich (inspirowanych Wikampem) w formie rywalizacyjnej gry planszowej symulującej postęp w nauce przez kolejne etapy studiów. Gra będzie nosić tytuł "Wyścig po Zaliczenie".

Zadanie 3 – Koncepcja Formy Gamifikacyjnej. Specyfikacja Projektu

Skład Zespołu i Tytuł Projektu:

Skład Zespołu:

- Bartłomiej Raj (251210)
- Marcel Podlecki (251204)
- Bartłomiej Jedyk (252795)
- Wojciech Stochmiątek (251226)
- Marcin Świstak (245944)

Tytuł Projektu:

- „Wyścig po Zaliczenie” (WpZ)

Opis Koncepcji, Zasad, Ról, Architektury, Logiki:

Koncepcja:

"Wyścig po Zaliczenie" to cyfrowa gra planszowa przeznaczona dla minimum dwóch graczy (studentów), której celem jest symulacja i uatrakcyjnienie procesu nauki i zaliczania kolejnych etapów studiów na Politechnice Łódzkiej. Gracze wcielają się w studentów-pionki, którzy ścigają się po planszy reprezentującej ścieżkę edukacyjną (np. pierwsze semestry), odpowiadając na pytania z różnych dziedzin wiedzy akademickiej. Gra ma na celu motywowanie do nauki poprzez rywalizację i zabawę.

Zasady Gry:

Plansza: Reprezentuje kolejne etapy nauki (np. tygodnie semestru, moduły przedmiotowe) które wybierze gracz, np. wybór semestru. Zawiera pola standardowe, pola "przedmiotowe" (aktywujące pytania), pola specjalne (bonusowe/karne).

Ruch: Gracze poruszają się po planszy zgodnie z wynikiem rzutu wirtualną, sześciościenną kostką.

Pola Przedmiotowe: Wylądowanie na takim polu skutkuje wyświetleniem pytania testowego (jednokrotnego lub wielokrotnego wyboru) z dziedziny odpowiadającej polu (pytania mogą być inspirowane bazami pytań z Wikampa lub stworzone na potrzeby gry).

Odpowiedzi: Poprawna odpowiedź może nagradzać gracza (np. dodatkowym ruchem, punktami, brakiem negatywnych konsekwencji). Błędna odpowiedź skutkuje karą (np. utratą kolejki, cofnięciem się na planszy, wzmocnieniem "Pionka Profesora").

"Pionek Profesora" (Opcjonalny Element Rywalizacji): Pionek sterowany przez komputer, który również porusza się po planszy (np. o stałą liczbę pól co turę lub przyspieszając po błędach graczy). Dodaje element presji czasu – gracze muszą ukończyć wyścig przed nim.

Pola Specjalne: Mogą oferować bonusy (np. "Konsultacje" - pominięcie pytania, "Projekt zespołowy" - dodatkowy ruch) lub kary (np. "Poprawka" - strata kolejki, "Nieoczekiwane kolokwium" - cofnięcie).

Cel Gry: Dotarcie do pola "Meta" (symbolizującego zaliczenie semestru/roku) jako pierwszy gracz lub przed "Pionkiem Profesora".

Role Użytkowników:

Student (Gracz): Aktywnie uczestniczy w grze, rzuca kostką, odpowiada na pytania, podejmuje decyzje strategiczne (jeśli mechanika na to pozwala), rywalizuje z innymi graczami i/lub "Pionkiem Profesora". Minimalnie dwóch graczy ludzkich.

Komputer (System Gry): Pełni rolę automatycznego systemu zarządzającego przebiegiem rozgrywki. Nie jest to rola "Mistrza Gry" ani aktywnego przeciwnika. Komputer odpowiada za: kontrolę tur, obsługę rzutów kostką, losowanie i prezentację pytań z bazy, weryfikację odpowiedzi graczy, przesuwanie pionków graczy oraz "Pionka Profesora" (zgodnie ze zdefiniowanymi zasadami jego ruchu), egzekwowanie reguł pól specjalnych, zarządzanie punktacją/rankingiem oraz ogłaszanie zwycięzcy/zakończenia gry. Działa wyłącznie jako egzekutor zaprogramowanych zasad.

Architektura i Logika:

Rdzeń Logiki Gry (Game Engine): Moduł odpowiedzialny za zarządzanie stanem gry (pozycje graczy, aktualna tura, stan "Profesora", punkty), implementację zasad ruchu, aktywacji pól, logiki pytań i odpowiedzi, warunków zwycięstwa/przegranej.

Baza Danych Pytań: Moduł przechowujący pytania, odpowiedzi, powiązania z polami/przedmiotami oraz ewentualnie poziom trudności. Może być zaimplementowana jako plik (JSON, CSV) lub prosta baza danych (np. SQLite) dla łatwości zarządzania.

Interfejs Użytkownika (GUI): Moduł odpowiedzialny za wizualną prezentację gry: wyświetlanie planszy, pionków, kostki, okien dialogowych z pytaniami, paneli informacyjnych (HUD) graczy, animacji. Obsługuje również interakcje ze strony gracza (kliknięcia przycisków, wybór odpowiedzi).

Moduł Sieciowy (Opcjonalnie, poza podstawowym zakresem): W przypadku implementacji gry sieciowej, moduł ten odpowiadałby za komunikację i synchronizację stanu gry między klientami. W wersji podstawowej gra będzie działać lokalnie na jednym ekranie.

Logika Przepływu Gry: Inicjalizacja gry (ustawienie graczy) -> Pętla główna (dopóki warunek końca gry nie jest spełniony): Określenie aktywnego gracza -> Akcja gracza (rzut kostką) -> Przesunięcie pionka -> Aktywacja logiki pola (standardowe, pytanie, specjalne) -> Obsługa akcji pola (wyświetlenie pytania, weryfikacja odpowiedzi, zastosowanie bonusu/kary) -> Aktualizacja stanu gry (pozycje, punkty) -> Ruch "Pionka Profesora" (jeśli aktywny) -> Sprawdzenie warunku końca gry -> Przejście do następnego gracza.

Zadanie 2: Przegląd Narzędzi do Prototypowania dla Projektu "Wyścig po Zaliczenie"

Celem tego zadania jest indywidualne zbadanie przez każdego członka zespołu trzech przydzielonych narzędzi do tworzenia cyfrowych makiet i prototypów. Analiza powinna skupić się na ocenie przydatności danego narzędzia do stworzenia prototypu gry "Wyścig po Zaliczenie", biorąc pod uwagę specyficzne wymagania projektu.

Kryteria Oceny (dla każdego narzędzia):

Dopasowanie do Koncepcji Gry:

Symulacja Planszy/Mapy: Czy narzędzie pozwala na łatwe stworzenie i manipulację widokiem przypominającym planszę do gry? Czy wspiera "kamerę" (np. zoom, przesuwanie widoku, jeśli plansza jest duża)?

Obsługa Grafiki: Czy umożliwia import i pracę z grafiką rastrową (np. tła, postacie) i wektorową (np. ikony, przyciski)?

Animacje: Czy wspiera tworzenie:

Mikroanimacji elementów UI (np. hover na przycisku, pojawianie się okna)?

Animacji ruchu obiektów (np. przesuwanie pionka, animacja kostki)?

Prostych animacji feedbacku (np. błysk po odpowiedzi)?

Interaktywność: Czy pozwala na tworzenie interaktywnych elementów (klikalne przyciski, wybór opcji)? Czy symuluje logikę warunkową (np. pokazanie innego ekranu/stanu w zależności od wybranej odpowiedzi)?

Obsługa Dźwięku: Czy możliwe jest dodanie i odtwarzanie dźwięków w prototypie (np. dźwięk kostki, odpowiedzi)? (Nawet podstawowa obsługa jest plusem).

Interakcja Dotykowa (Gesty): Czy wspiera gesty typowe dla ekranów dotykowych? (Mniej krytyczne dla gry planszowej na desktop, ale warto sprawdzić).

Zdarzenia Czasowe/Losowe: Czy umożliwia symulację zdarzeń wyzwalanych czasowo (np. limit na odpowiedź, ruch "Profesora") lub losowo (np. losowe wydarzenie na polu specjalnym)?

Ograniczenia Praktyczne:

System Operacyjny: Czy działa na różnych systemach (Windows, macOS, Linux, Web)? Czy są jakieś ograniczenia?

Licencja/Koszt: Jaki jest model licencjonowania? Czy istnieje darmowa wersja/plan lub okres próbny (trial)? Czy jest on wystarczająco długi/funkcjonalny do stworzenia prototypu na potrzeby projektu?

Praca Zespołowa: Czy narzędzie umożliwia współpracę w czasie rzeczywistym lub łatwe łączenie pracy kilku osób nad jednym projektem?

Limity: Czy darmowa wersja/trial ma ograniczenia (np. liczba ekranów, projektów, funkcji), które mogłyby uniemożliwić stworzenie prototypu gry?

Przydział Narzędzi do Analizy:

Bartłomiej Raj:

Figma: Popularne, webowe narzędzie UI/UX z mocnym darmowym planem. Znane z łatwości współpracy i bogatej biblioteki wtyczek. Należy sprawdzić głębokość symulacji interakcji i animacji pod kątem gry.

Balsamiq: Skupione na szybkim tworzeniu makiet niskiej wierności (low-fidelity). Łatwe do nauczenia. Trzeba ocenić, czy wystarczy do pokazania przepływu gry, mimo ograniczeń w wizualizacjach i animacjach.

MarvelApp: Narzędzie webowe, często postrzegane jako proste w obsłudze, dobre do tworzenia klikalnych prototypów z gotowych ekranów. Zbadać możliwości animacji i interakcji.

Marcel Podlecki:

Adobe XD: Główny konkurent Figmy i Sketch, dostępny na Win/Mac. Sprawdzić możliwości prototypowania, animacji (auto-animate) i ewentualne ograniczenia darmowego planu.

Proto.io: Webowe narzędzie znane z zaawansowanych możliwości tworzenia interakcji i animacji, często używane do mobilnych prototypów. Ocenić krzywą uczenia i dopasowanie do gry planszowej.

Whimsical: Narzędzie webowe, silne w tworzeniu diagramów przepływu, map myśli i prostych wireframe'ów. Ocenić, jak jego funkcje mogą wesprzeć prototypowanie logiki i struktury gry.

Bartłomiej Jedyk:

Sketch: Bardzo popularne narzędzie UI/UX, ale dostępne tylko na macOS. Ważne ograniczenie do odnotowania. Silne w projektowaniu wektorowym i systemach designu. Sprawdzić możliwości prototypowania.

InVision (Studio / Web): Ekosystem narzędzi do prototypowania i współpracy. InVision Studio (desktop) oferuje zaawansowane animacje. Zbadać różnice między wersją webową a Studio i model licencjonowania.

Moqups: Webowe narzędzie do tworzenia wireframe'ów, makiet i diagramów. Ocenić zakres funkcji prototypowania i interaktywności.

Wojciech Stochmiątek:

Framer: Narzędzie webowe, które ewoluowało w kierunku silnej integracji z kodem (React). Pozwala na tworzenie bardzo zaawansowanych, kodowych prototypów. Ocenić próg wejścia i czy taka moc jest potrzebna/przydatna dla tego projektu.

Axure RP: Desktopowe narzędzie (Win/Mac) znane z możliwości tworzenia bardzo złożonych, dynamicznych prototypów z logiką warunkową i zmiennymi. Uważane za mające stromą krzywą uczenia.

Justinmind: Desktopowe narzędzie (Win/Mac) oferujące szeroki zakres funkcji do tworzenia interaktywnych prototypów, w tym dla aplikacji webowych i mobilnych. Sprawdzić łatwość użycia i możliwości symulacji.

Marcin Świstak:

UXPin: Narzędzie skupiające się na interaktywnych stanach, zmiennych i logice warunkowej, z możliwością importu z innych narzędzi (Sketch, Figma). Ocenić, jak te funkcje mogą pomóc w symulacji gry.

Affinity Designer: Profesjonalne oprogramowanie do grafiki wektorowej (konkurent Adobe Illustratora), dostępne na Win/Mac/iPad, z jednorazową opłatą. Posiada pewne funkcje do projektowania UI. Należy dokładnie sprawdzić jego (prawdopodobnie ograniczone) możliwości prototypowania interakcji i animacji.

Mockplus: Platforma oferująca narzędzia do prototypowania (desktop i web). Zbadać zakres funkcji, łatwość użycia i model cenowy/darmowy.